

## 중간고사

1.  $h(n) = n \quad f(n) = \Theta(n \log^2 n)$

$$\frac{f(n)}{h(n)} = \Theta(\log^2 n) \text{ 이면}$$

$$\therefore T(n) = \Theta(n \log^3 n)$$

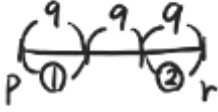
2.

for  $i \leftarrow p$  to  $r$

Sum  $\leftarrow$  Sum + A[i]

↳ A 배열의 인덱스 p에서 r까지 합을 구하므로 작업 크기는 n

$$tmp \leftarrow \text{sum} + S(A, p, p+q-1) + S(A, p+2q, r)$$

↳   $q = \frac{r-p+1}{3}$  이므로  
①과 ②는 각각  $T(\frac{n}{3})$ 의 작업

$$T(n) = 2T(\frac{n}{3}) + n$$

$$h(n) = n^{\log_3 2} \quad f(n) = n$$

$$\frac{f(n)}{h(n)} = \frac{n^{\text{하항}}}{n^{\log_3 2}} = \Omega(n^\epsilon) \quad (\epsilon \text{는 임의의 양수}) \text{ 이고}$$

어떤 상수  $C (< 1)$ 와 충분히 큰 모든  $n$ 에 대해

$$\frac{2}{3}n \leq Cn \text{ 이므로}$$

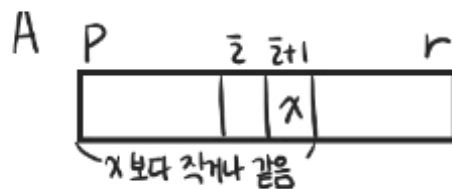
$$\therefore T(n) = \Theta(n) \text{ 이다.}$$

3. Partition 함수는 j를 증가시키면서 A[j]가 x보다 작은 경우

i를 증가시키고 A[i]  $\leftrightarrow$  A[j] swap 한 뒤  $i \leftarrow i+1$ 을 해야 한다.

이렇게 수정한 코드에서 i+1은 배열에서 x보다 작거나 같은 원소들의 끝 인덱스가

된다.



정답 :

```

partition(A[], p, r)
{
    x ← A[r];
    i ← p-1;
    for j ← p to r-1
        if (A[j] ≤ x) then (A[i+1] ↔ A[j];  $\bar{z} \leftarrow \bar{z}+1$ ;)
    A[i+1] ↔ A[r];
    return i+1;
}

```

4.  $-k \sim k$ 를 C[]에 대응하면

실제 값	$-k$	$-(k-1)$	$\dots$	$0$	$\dots$	$k$
인덱스	$0$	$1$		$k$		$2k$

$\bar{z}$ 의 범위는  $0 \sim 2k$

$C[A[j]+k]$ : A[j] 값을 가진 원소의 개수로 수정

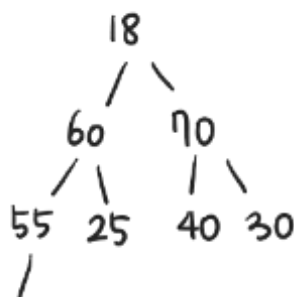
정답 :

```

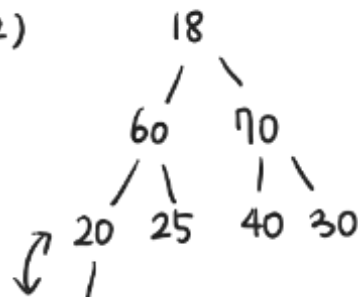
countingSort(A[], B[], n)
{
    for  $i \leftarrow 1$  to  $k$   $\bar{z} \leftarrow 0$  to  $2k$ 
        C[i] ← 0;
    for j ← 1 to n
         $C[A[j]] \leftarrow C[A[j]] + 1$   $C[A[j]+k] \leftarrow C[A[j]+k] + 1$ ;
    for  $i \leftarrow 2$  to  $k$   $\bar{z} \leftarrow 1$  to  $2k$ 
        C[i] ← C[i] + C[i-1];
    for j ← n downto 1 {
         $B[C[A[j]]] \leftarrow A[j]$   $B[C[A[j]+k]] \leftarrow A[j]$ ;
         $C[A[j]] \leftarrow C[A[j]] - 1$   $C[A[j]+k] \leftarrow C[A[j]+k] - 1$ ;
    }
}

```

5. 1) 시작

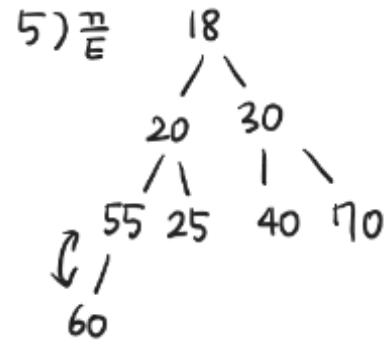
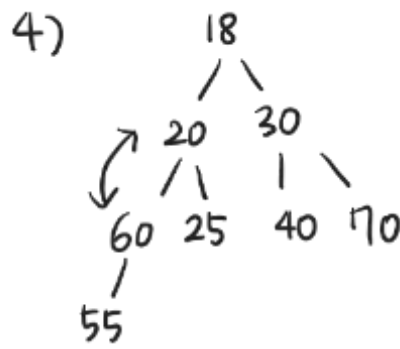
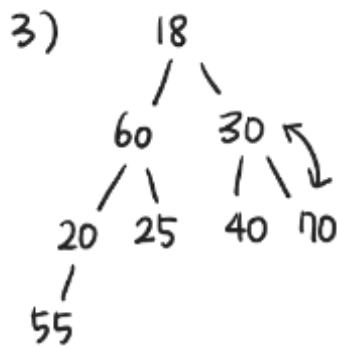


2)



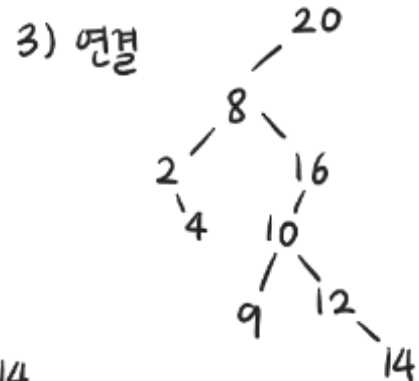
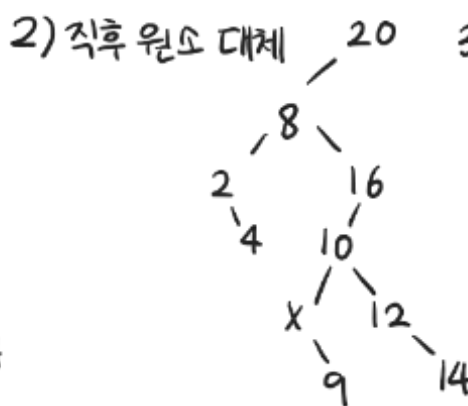
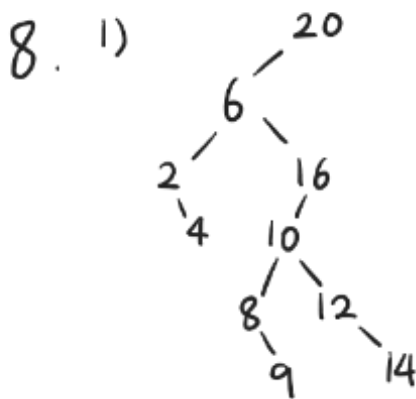
20

55

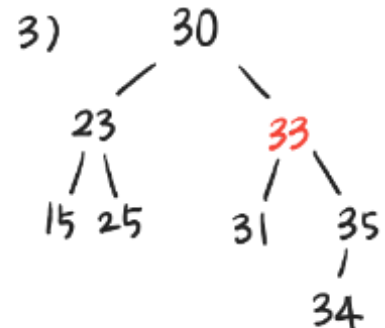
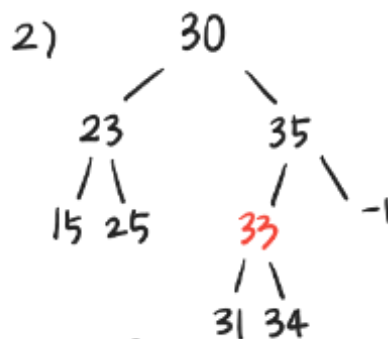
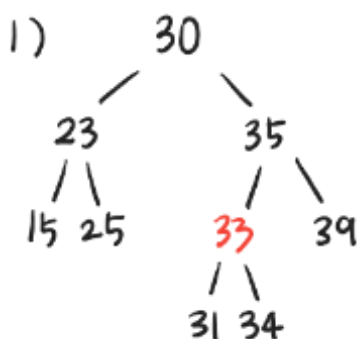


6. ① 삽입정렬 선형시간이 걸립니다. 정렬된 배열에 비교하면서 삽입하는 과정으로 삽입할 원소가 정렬된 배열의 원소보다 작으면 배열의 위치 변화가 필요하여 SWAP이 이뤄지지만 첫 비교부터 정렬된 배열의 원소보다 삽입할 원소가 더 크기 때문에 비교만 이뤄지고 정렬이 끝난다.

7. 기속정렬에서 <sup>현재</sup> 정렬기준으로 정한 자릿 수가 같더라도 그 이전 자릿수값에 따라 크기가 정렬이 되어있기 때문에 안정성 지원여부가 정렬정확도에 영향을 준다.



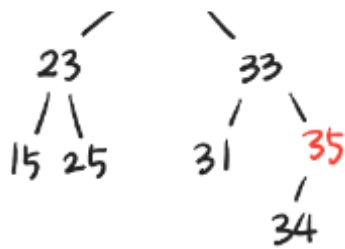
9.



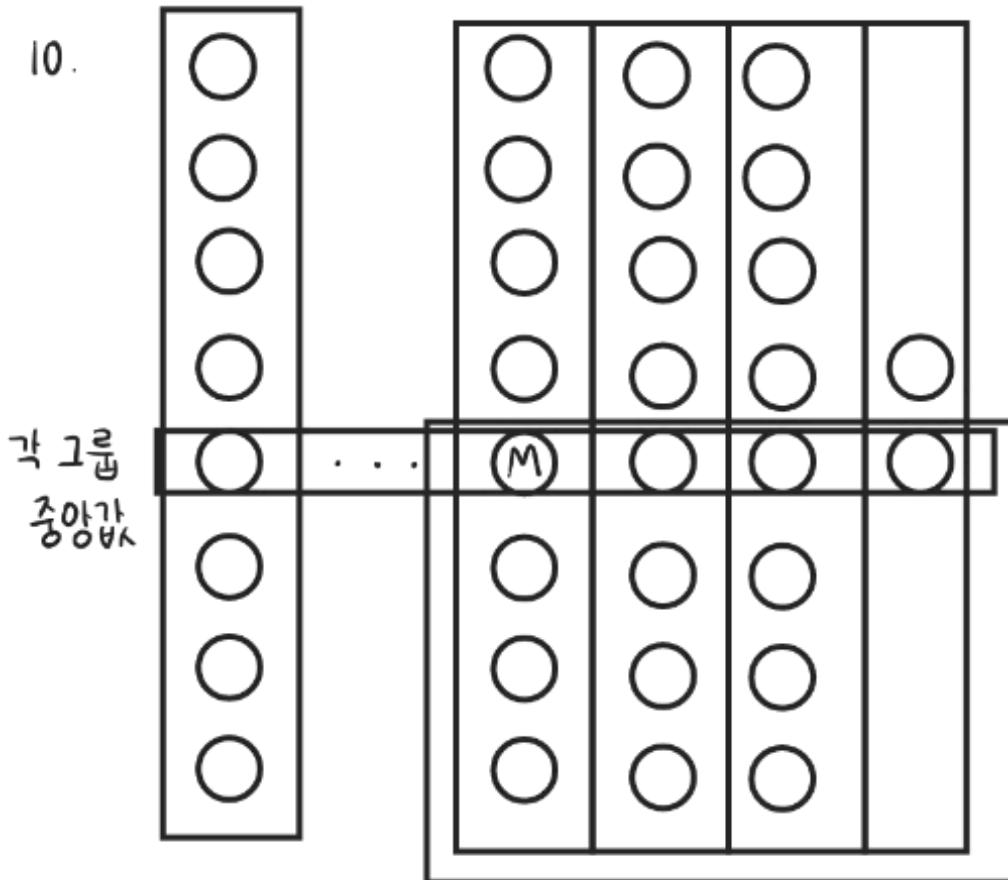
< 규칙 위반  
재정렬 필요!! >

< 35를 중심으로 오른쪽 회전 >

4) 30



< 33과 35 색상 변경 >



M보다 큰 원소는 최소

$\frac{n}{8}$  그룹  $\times \frac{1}{2} \times 4$  개 - 3 개 있어야 하므로

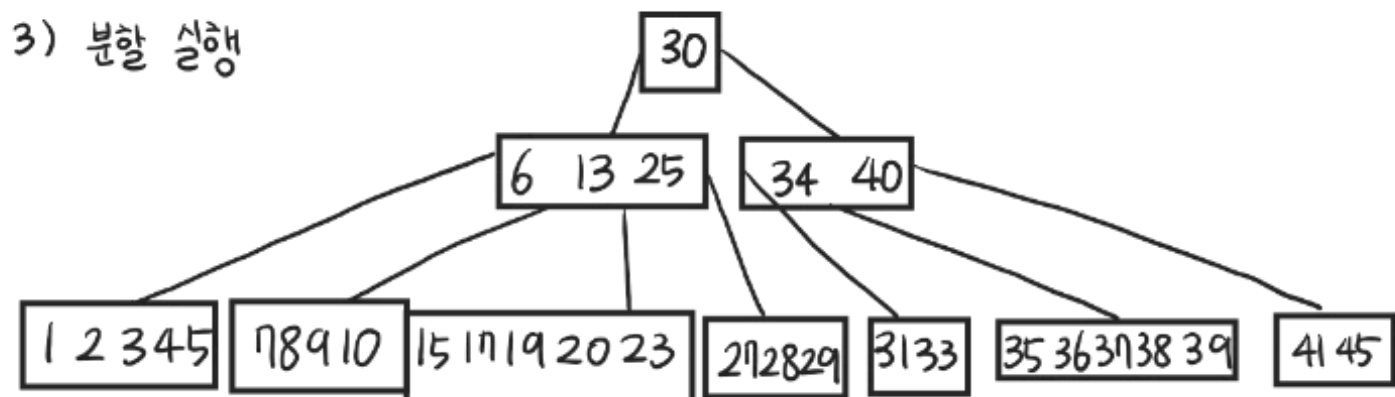
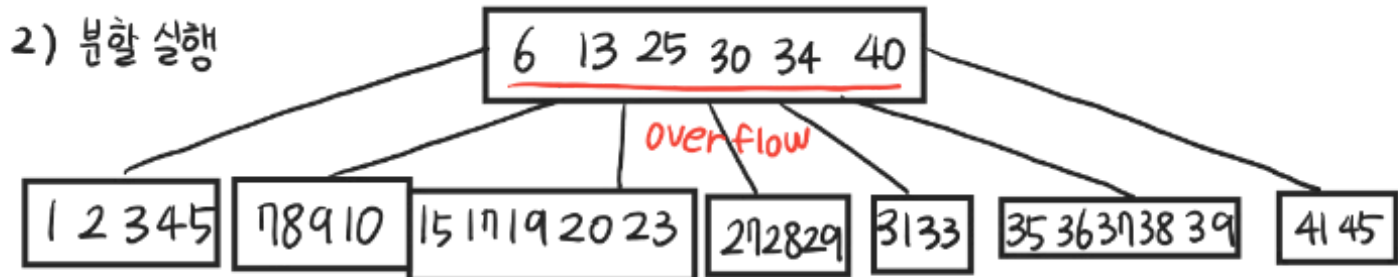
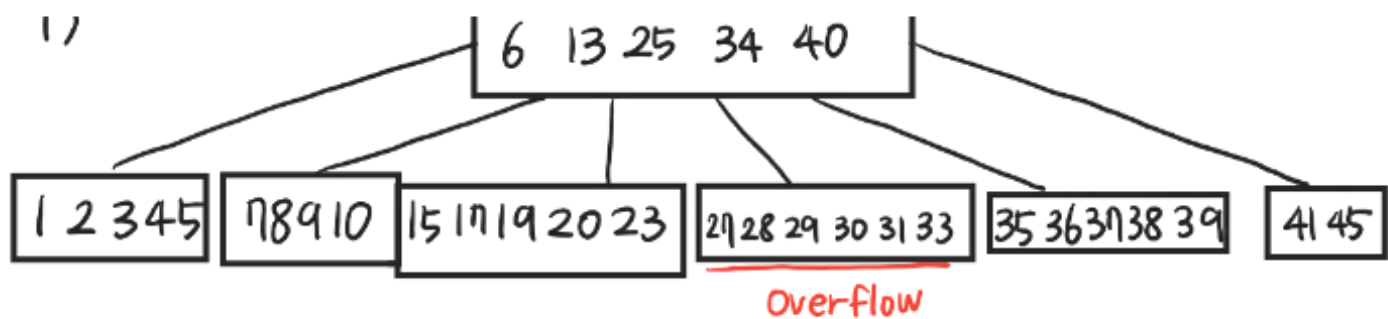
최악의 경우 분할 비율은

$$\frac{3}{4}n+3 : \frac{n}{4}-3 \text{ 이다.}$$

11. LinearSelect 알고리즘을 이용해 선형시간에 찾은 중앙값을 기준으로 배열을 절반으로 분할하면  $T(n) = 2T(\frac{n}{2}) + \Theta(n)$  이 되고 최악의 경우에도  $T(n) = \Theta(n \log n)$  가 된다.

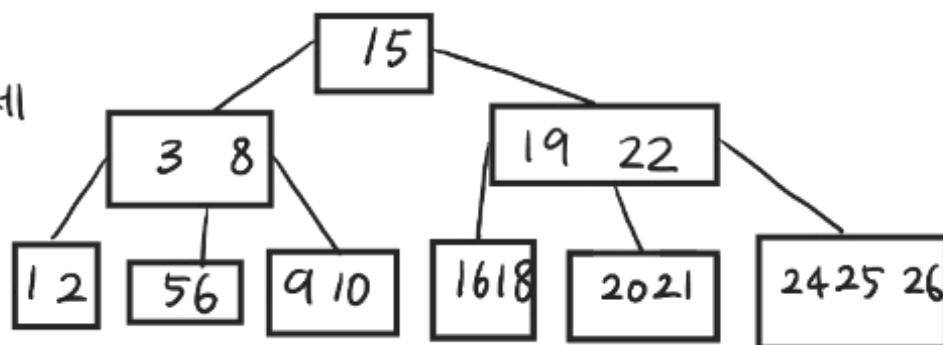
12.

13

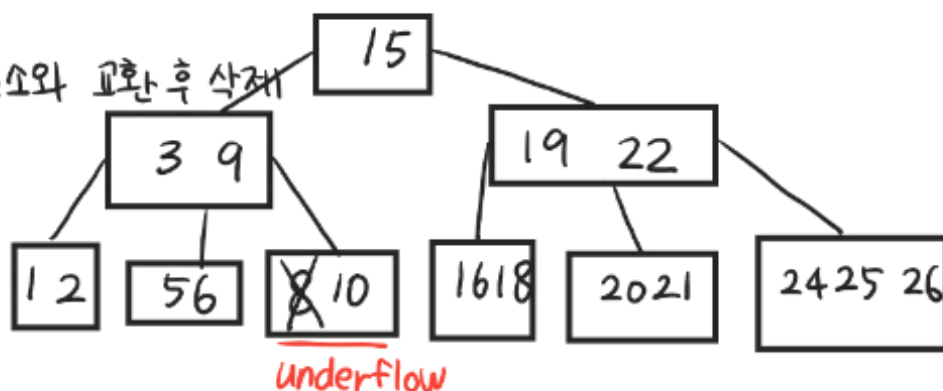


13.

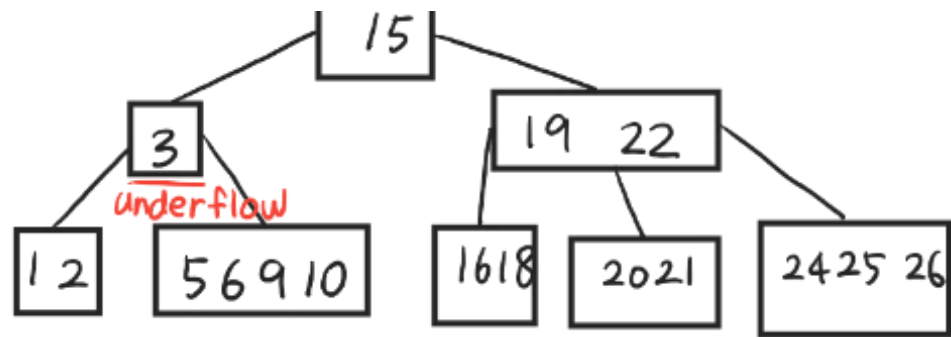
1-1) 8삭제



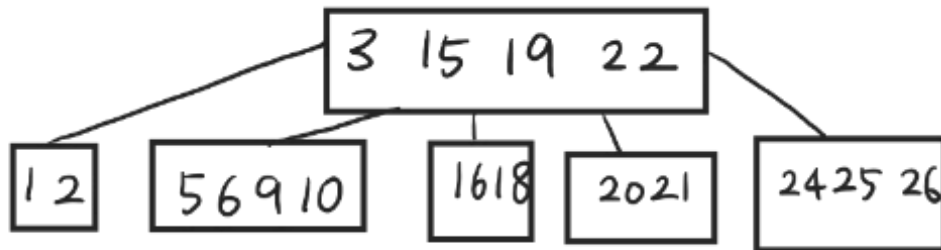
1-2) 직후원소와 교환 후 삭제



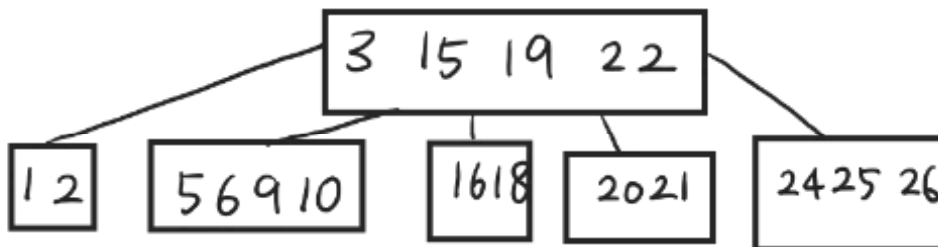
1-3) 병합



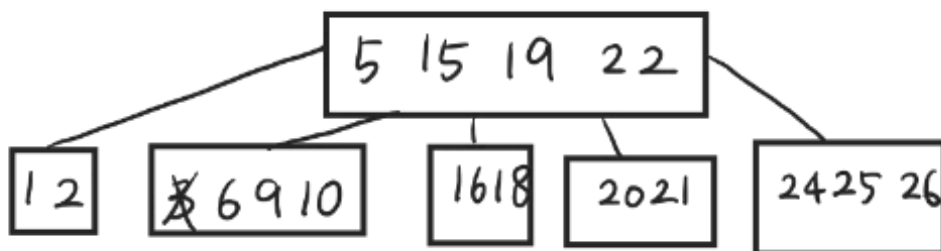
1-4) 병합



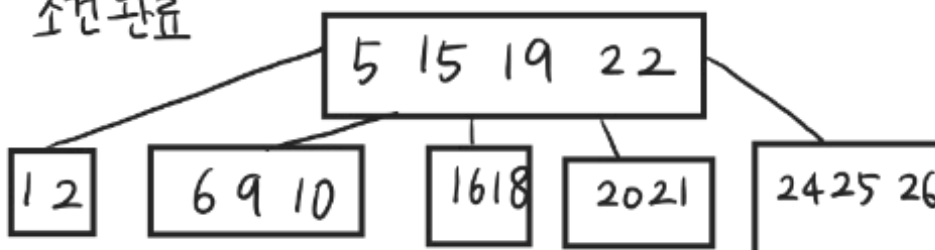
2-1) 3 삭제



2-2) 직후요소와 교환 후 삭제



2-3) 조건완료



14.

address data  
0 100

조사횟수  
1

5

조사 회수

0 1 1 1 1 1

9 1000 10

10 101 11 -1

⋮ ⋮

19 1001 20 -1

20 102 21 -2

⋮ ⋮

29 1002 30 -2

30 103 31 -3

⋮ ⋮

39 1003 40 -3

40 104 41 -4

⋮ ⋮

49 1004 50 -4

$$\sum_{k=1}^{46} k + 10 + 19 + 28 + 37$$

$$= \frac{46 \times 47}{2} + 94$$

$\text{평균} \frac{1175}{50} = 23.5 \text{ 번}$
--